

## Pemodelan Curah Hujan-Limpasan Menggunakan Artificial Neural Network (ANN) dengan Metode Backpropagation

Iwan K. Hadihardaja<sup>1)</sup>  
Sugeng Sutikno<sup>2)</sup>

### Abstrak

Hubungan curah hujan-limpasan telah dikembangkan secara terus-menerus dengan menerapkan artificial intelligence sebagai alternatif pemodelan yang pada penelitian ini menggunakan black box model yakni Jaringan Syaraf tiruan (artificial neural network). Dengan menggunakan model kotak hitam (black box model) tersebut, maka dalam penerapannya tidak membutuhkan pengetahuan yang kompleks antar elemen-elemen dalam suatu sistem Daerah Aliran Sungai (DAS) yang juga tidak secara eksplisit merepresentasikan hubungan antar elemen dalam DAS dan proses interaksi curah hujan-limpasan. Sehingga perubahan antar elemen di dalam suatu DAS tidak perlu diketahui sepanjang hujan dan limpasan diukur dengan akurat dari waktu ke waktu, oleh karena itu pemodelan dapat dilakukan secara lebih sederhana dengan hanya memiliki input hujan dan output limpasan didalam pengembangan model tersebut. Studi kasus yang digunakan adalah data curah hujan tengah bulanan dan debit limpasan DAS Way Sekampung – Pujorahayu, selama kurun waktu 19 tahun. Data curah hujan merupakan input sedangkan data debit merupakan variable output. Dari hasil penelitian ini diperoleh koefisien korelasi tertinggi sebesar 0,813 atau 81,3%. Dalam studi ini disimpulkan bahwa secara umum ANN dapat diterapkan dalam pemodelan curah hujan-limpasan, walaupun hasilnya belum terlalu akurat oleh karena masih ada penyimpangan.

**Kata-kata Kunci :** Pemodelan hujan-limpasan, jaringan syaraf tiruan, metode backpropagation.

### Abstract

Rainfall-runoff relation has been developed continuously by applying artificial intelligence as one of the black box model alternative called Artificial Neural Network. By applying black box model, it is not necessary to apply complexity of knowledge due to interrelated elements in a river basin in which it is not explicitly representing the relation of the elements and interaction process of the rainfall-runoff modeling. Consequently, the changes of the elements in a river basin is not necessary to be quantified as long as rainfall dan runoff is observed accurately from time to time, furthermore, the modeling can be applied within less complexity due to rainfall and runoff data observation as an input and output, respectively. The case study applied to the river flow on the way Sekampung River in Lampung Province. The data used is rainfall data and stream flow discharge data in the middle of the month on the water level station Pujorahayu, for 19 years from 1983 up to 2001. The rainfall data is input and stream flow is a variable output. Learning method that is used reduced gradient. From the result of this research got correlation coefficient 0,813 or 81,3% the tallest. The conclusion of this research is the generally ANN can implemented in the rainfall run off modeling, although the result is not extremely accurate yet because of the deviation between observation and result of the model.

**Keywords :** Rainfall-runoff modeling, artificial neural network, backpropagation method.

### 1. Pendahuluan

Perencanaan, pengelolaan dan pengembangan sumberdaya air selalu memerlukan analisa terhadap variabel hidrologi seperti curah hujan, dan aliran sungai. Untuk keperluan analisa hidrologi diperlukan data hidrologi yang panjang, tetapi sering dijumpai data yang tersedia tidak lengkap atau bahkan tidak ada

sama sekali. Sesuai dengan karakteristik fenomena hidrologi suatu daerah pengaliran sungai, aliran sungai berubah-ubah tidak beraturan, oleh karena itu sukar untuk meramalkan besarnya debit yang melintasi penampang sungai secara pasti pada suatu saat tertentu. Untuk mendekati fenomena tersebut maka perlu dikembangkan suatu analisa sistem hidrologi dengan menggunakan model yang merupakan

1. Staf Pengajar Departemen Teknik Sipil, FTSP-ITB.

2. Staf pengajar Jurusan Teknik Sipil, Fakultas Teknik-Universitas Subang.

**Catatan :** Usulan makalah dikirimkan pada 13 Mei 2005 dan dinilai oleh peer reviewer pada tanggal 16 Mei 2005 - 13 Juni 2005. Revisi penulisan dilakukan antara tanggal 15 Juni 2005 hingga 20 September 2005.

penyederhanaan kenyataan alam yang sebenarnya.

Salah satu alternative model yang dapat dikembangkan untuk pemodelan hidrologi adalah dengan menggunakan jaringan syaraf tiruan (artificial neural network). *Artificial neural network (ANN)* yang merupakan salah satu bentuk kecerdasan buatan telah banyak diaplikasikan dalam bidang computer, teknik listrik, fisika, *financial*, dan lain-lain. Penerapan ANN dalam bidang sumberdaya air antara lain telah diterapkan untuk mencari hubungan limpasan dan sedimen [Hadihardaja, 2002], pemodelan *multivariate* deret waktu [Saragih, Hadihardaja, 2003], dan model hujan-limpasan [Hadihardaja, Sugeng, 2005]. Ketiga model yang telah dikembangkan tersebut sama-sama menggunakan metode Generalized *Reduced Gradient* (GRG). Pada penelitian ini pemodelan curah-hujan limpasan menggunakan ANN dengan metode *Backpropagation*.

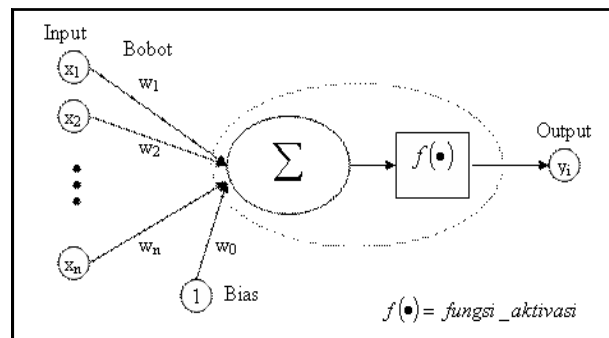
## 2. Lokasi Studi

Lokasi studi adalah daerah aliran sungai Way Sekampung yang berada di Propinsi Lampung. Data debit aliran sungai yang digunakan sebagai data target pada pemodelan diambil pada pos duga air Pujorahayu pada sungai Way Sekampung dengan luas daerah pengaliran 1696 km<sup>2</sup>. Pos duga air Pujorahayu terletak pada 05°19'35" LS, 105°03'39" BT dengan elevasi pos duga air + 83 meter di atas muka air laut. Jenis alat adalah pesawat pos duga air mingguan didirikan pada tanggal 1 Mei 1968. Lokasi pos duga air Pujorahayu adalah di desa Pujorahayu, kecamatan Gedong Tataan, kabupaten Lampung Selatan propinsi Lampung. Data curah hujan yang digunakan sebagai data masukan diperoleh dari stasiun pengamat hujan yang berada dalam wilayah DAS Way Sekampung-Pujorahayu sebanyak sembilan stasiun pencatat hujan. Pada penelitian ini di gunakan data curah hujan dan data debit tengah bulanan selama 19 tahun dari tahun 1983 sampai tahun 2001.

## 3. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (ANN) merupakan salah satu representasi tiruan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Otak manusia berisi berjuta-juta sel syaraf (neuron) yang bertugas untuk memproses informasi. Masing-masing sel saling berinteraksi sehingga mendukung kemampuan kerja otak. Setiap sel syaraf akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi.

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, yang ada hubungan antara neuron-neuron tersebut. Neuron-neuron akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju neuron-neuron yang



Gambar 1. Struktur neuron jaringan syaraf

lain. Dalam jaringan syaraf hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Struktur neuron pada jaringan syaraf ditunjukkan pada **Gambar 1**.

Neuron-neuron buatan tersebut bekerja dengan cara yang sama dengan neuron-neuron biologis. Informasi atau masukan (*input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tetapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron diaktifkan, maka neuron akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya, demikian seterusnya.

Secara umum jaringan syaraf tiruan terdiri dari 3 (tiga) lapisan yaitu :

- *Input layer*, terdiri dari neuron-neuron yang menerima sebuah input dari lingkungan luar. Neuron tersebut tidak melakukan suatu perubahan dalam masukan, tetapi hanya mengirimkannya ke elemen pemroses (*processing elements*) yang berdekatan dengan lapisan tersembunyi (*hidden layer*).
- *Hidden layer*, terdiri dari neuron-neuron yang mempunyai tipe menerima masukan yang dipindahkan dari input layer, melakukan perubahan di dalamnya, dan melemparkan output ke lapisan berikutnya.
- *Output layer*, terdiri dari neuron-neuron yang menerima output dari *hidden layer* dan mengirimkannya kepada pemakai.

## 4. Pemodelan Jaringan Syaraf Tiruan

Prosedur pemodelan dengan ANN secara umum terdiri dari empat langkah yaitu, penyiapan data,

pemilihan arsitektur, pembelajaran (*learning, taining*), pengujian (*testing*). Untuk menilai kedekatan atau kecocokan data hasil pemodelan dengan data hasil pengamatan, dilakukan uji kecocokan dengan menggunakan fungsi objektif atau fungsi kesalahan yang merupakan persamaan dari perhitungan dan pengamatan. Bentuk dari fungsi objektif ini bermacam-macam, dalam penelitian ini digunakan kesalahan absolute rata-rata (KAR). Persamaan untuk menghitung mengitung KAR menggunakan persamaan yang di formulasikan oleh Adidarma, Hadihardaja dan Legowo at al. [2004] yaitu :

Kesalahan Absolut Rata-Rata (KAR) =

$$\frac{1}{n} \sum \frac{Abs(Q_{comp} - Q_{obs})}{Q_{obs}}$$

dimana :

$Q_{comp}$  = debit hasil perhitungan

$Q_{obs}$  = debit hasil pengamatan

N = banyak data

Penerapan Kesalahan absolut rata-rata (KAR) dalam penelitian ini karena menurut Adidarma, Hadihardaja dan Legowo [2004], Nilai KAR lebih memperhatikan debit kecil daripada debit besar sehingga sangat bermanfaat bagi pengembangan sumberdaya air. Fungsi objektif seperti KAR dilakukan dengan pertimbangan penyimpangan terhadap data pengamatan menjadi pusat perhatian sehingga debit rendah menjadi lebih berarti.

## 5. Menyiapkan Data

Sebelum memulai pembelajaran disiapkan tiga bagian data yang akan digunakan sebagai data pembelajaran (*training*), data pengujian (*testing*), dan data validasi. Data training digunakan data tahun 1983 sampai 1992. Data testing digunakan data tahun 1995 sampai tahun 2001. Data validasi digunakan data tahun 1994 sampai tahun 1995.

Data input dan output dalam jaringan *backpropagation* berada dalam batas asimtot 0 (nol) sampai 1 (satu), sehingga data curah hujan dan data debit aliran sungai yang akan digunakan sebagai data input dan data target, terlebih dahulu harus diskala atau dinormalkan sehingga berada dalam batas 0 (nol) sampai 1 (satu).

Normalisasi data menggunakan persamaan seperti dalam Babovic dan Gopakumar at al. [1999] sebagai berikut :

Normalisasi dari data asli X menjadi X' :

$$X' = \alpha X + \beta$$

$$\alpha = \frac{Upperbound - Lowerbound}{Max(X) - Min(X)}$$

$$\beta = Upperbound - (\alpha * Max(X))$$

dimana :

X' = data yang diskala

X = data asli

Upperbound = batas atas nilai data pada ANN = 1

Lowerbound = batas bawah nilai data pada ANN = 0

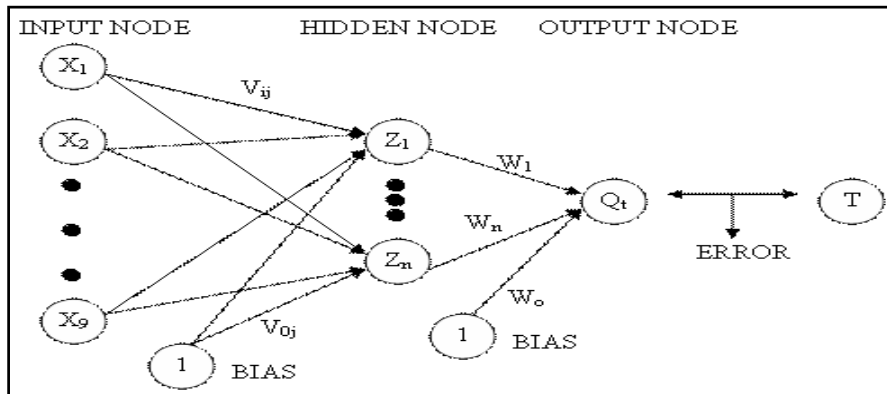
Untuk merubah kembali dari data yang diskala menjadi data asli menggunakan persamaan sebagai berikut

$$X = \frac{(X' - \beta)}{\alpha}$$

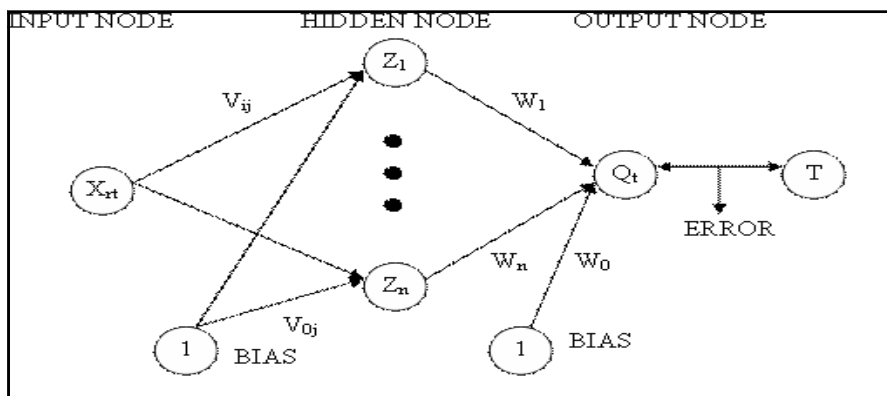
## 6. Arsitektur Jaringan

Pengaturan dari saraf-saraf dalam lapisan (*layers*) dan pola hubungannya diantara lapisan-lapisan disebut arsitektur jaringan. Berdasarkan jumlah lapisannya, ANN dapat diklasifikasikan sebagai jaringan dengan lapisan tunggal (*single layer net*) dan jaringan dengan banyak lapisan (*multi layer net*). Menurut Kusumadewi at al. [2003], jaringan dengan banyak lapisan dapat menyelesaikan permasalahan yang lebih sulit dari pada lapisan tunggal, tentu saja dengan pembelajaran yang lebih sulit. Namun demikian banyak kasus pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah. Arsitektur ANN di tentukan oleh lapisan tersembunyi (*hidden layer*) dan jumlah syaraf (*node*) pada setiap lapisan. Pada penelitian ini dipakai satu lapisan tersembunyi.

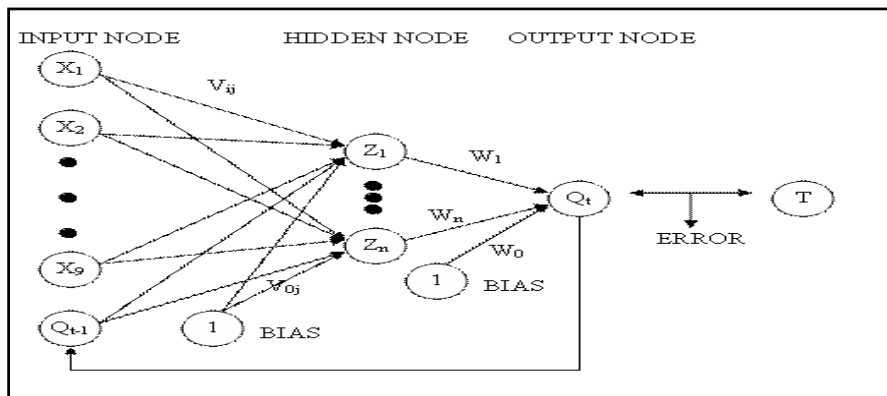
Berdasarkan syaraf masukan arsitektur jaringan pada penelitian ini tentukan menjadi empat arsitektur jaringan. Arsitektur 1 memiliki 9 syaraf masukan, ditentukan dari jumlah stasiun pengamat hujan dalam DAS. Arsitektur 2 memiliki 1 syaraf masukan, ditentukan dari hujan rata-rata dalam DAS. Arsitektur 3 memiliki 10 syaraf masukan yaitu 9 dari jumlah stasiun pengamat hujan dan 1 dari debit aliran sungai untuk periode sebelumnya. Arsitektur 4 memiliki 2 syaraf masukan yaitu 1 dari curah hujan rata-rata, dan 1 dari debit aliran sungai periode sebelumnya. Jumlah syaraf pada lapisan tersembunyi dipakai 4 syaraf dan 6 syaraf. Jumlah syaraf pada lapisan keluaran (*output layer*) dalam penelitian ini adalah satu syaraf yaitu debit aliran sungai. Secara keseluruhan arsitektur ANN yang di coba pada penelitian ini berdasarkan syaraf masukan (*input node*) di kelompokkan menjadi empat Arsitektur. Variasi arsitektur ANN dalam dalam penelitian ini disajikan pada **Gambar 2** sampai **5**.



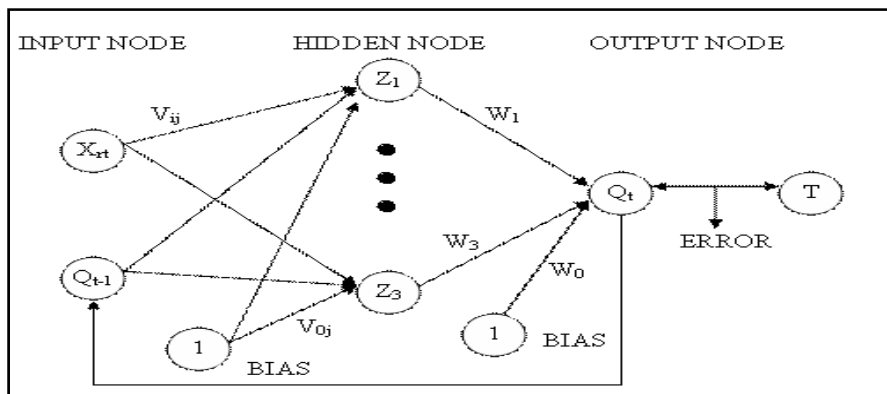
Gambar 2. Arsitektur 1



Gambar 3. Arsitektur 2



Gambar 4. Arsitektur 3



Gambar 5. Arsitektur 4

## 7. Fungsi Aktivasi

Fungsi aktivasi (*activation functions*) berguna untuk memeras fungsi output supaya berada dalam nilai ambang (*threshold*) yang diinginkan, yaitu berada antara nilai 0 sampai 1 atau -1 sampai 1. Dalam penelitian ini digunakan fungsi aktivasi sigmoid atau fungsi logistik (*logistic functions*). Fungsi sigmoid ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid memiliki nilai pada rentang 0 (nol) sampai 1 (satu). Fungsi *sigmoid* dirumuskan sebagai :

$$y = f(x) = \frac{1}{1 + e^{-\alpha x}}$$

## 8. Pembelajaran Metode *Backpropagation*

Proses pembelajaran dalam ANN bertujuan untuk mengubah faktor bobot sehingga diperoleh bobot hubungan yang diinginkan. Algoritma *backpropagation* menggunakan *error output* untuk mengubah bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid. Pada awal proses pembelajaran bobot hubungan diambil secara random. Algoritma pembelajaran, memodifikasi bobot tersebut dalam setiap epoch sampai pembelajaran selesai. Bila proses telah mencapai maksimum epoch yang direncanakan atau telah mencapai nilai error yang direncanakan maka proses berhenti. Koleksi bobot hubungan yang tersimpan digunakan sebagai bobot hubungan pada proses pengujian.

Proses pembelajaran *backpropagation* dimulai dengan menentukan target error, maksimum epoch, *learning rate* ( $\alpha$ ) dan memasukkan bobot awal secara acak ke dalam model. Algoritma pembelajaran metode *backpropagation* adalah sebagai berikut :

1. Inisialisasi bobot awal (ambil bobot awal dengan nilai random yang kecil)
2. Lakukan tahap perambatan maju (*forward propagation*) untuk mendapatkan error :
  - a. Tiap unit input ( $X_i$ ,  $i = 1, 2, 3, \dots, n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
  - b. Tiap-tiap unit tersembunyi ( $Z_i$ ,  $i = 1, 2, 3, \dots, p$ ) menjumlahkan sinyal-sinyal input terbobot ;

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:  $z_j = f(z\_in_j)$  dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit outputnya).

- c. Tiap-tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, k$ ), menjumlahkan sinyal-sinyal input terbobot.

$$y\_in = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya :  $y = f(y\_in)$ , dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit output).

3. Lakukan tahap perambatan mundur (*backward propagation*)

- a. Tiap-tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya :  $\delta_k = (t_k - y_k) f'(y\_in_k)$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ )

$$\Delta w_{jk} = \alpha \delta_k \cdot z_j$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$  :

$$\Delta w_{0k} = \alpha \delta_k$$

kirimkan  $\delta_k$  ini ke unit-unit yang ada di lapisan bawahnya.

- b. Tiap unit tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada dilapisan atasnya):

$$\delta\_in_j = \sum_{k=1}^p \delta_k \cdot w_{jk}$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta\_in_j f'(z\_in_j)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ )

$$\Delta v_{ij} = \alpha \delta_j x_i$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{0j}$ ):

$$\Delta v_{0j} = \alpha \delta_j$$

- c. Tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) memperbaiki bias dan bobotnya ( $j = 1, 2, 3, \dots, p$ ):  $w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$

Tiap-tiap unit tersembunyi ( $Z_j$ ,  $j=1, 2, 3, \dots, p$ ) memperbaiki bias dan bobotnya ( $i = 1, 2, 3, \dots, n$ ):  $v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$

Bobot-bobot baru ini digunakan sebagai bobot awal pada proses epoch berikutnya, proses ini dilakukan sampai target error tercapai atau sampai maksimum epoch. Untuk melakukan pembelajaran backpropagation dibuat program komputer dengan bahasa pemrograman *Borland Delphi 5* yang dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lain yang berbasis *MS-Windows*.

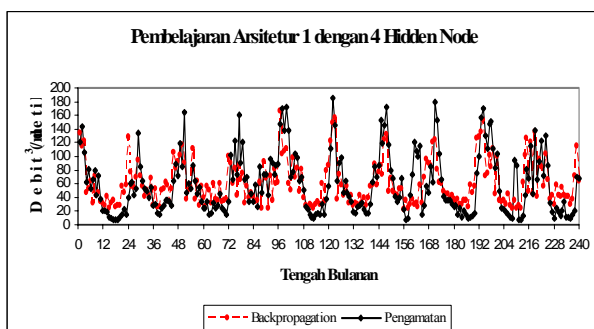
## 9. Analisa dan Diskusi

Dari hasil pembelajaran untuk masing-masing pola arsitektur dihitung parameter statistika yaitu kesalahan absolute rata-rata (KAR), standar deviasi dan koefisien korelasi. Parameter statistika untuk arsitektur 1 dengan 4 *hidden node* (A1-4), Arsitektur 1 dengan 6 *hidden node* (A1-6), arsitektur 2 dengan 4 *hidden node* (A2-4), arsitektur 2 dengan 6 *hidden node* (A2-6), arsitektur 3 dengan 4 *hidden node* (A3-4), arsitektur 3 dengan 6 *hidden node* (A3-6), arsitektur 4 dengan 4 *hidden node* (A4-4), dan arsitektur 4 dengan 6 *hidden node* (A4-6) disajikan pada **Table 1**. Berdasarkan parameter statistika menunjukan bahwa arsitektur 4 dengan 4 lapisan tersembunyi memiliki kesalahan absolute rata-rata paling kecil dan koefisien korelasi paling besar dari semua arsitektur yang sudah dilakukan pembelajaran.

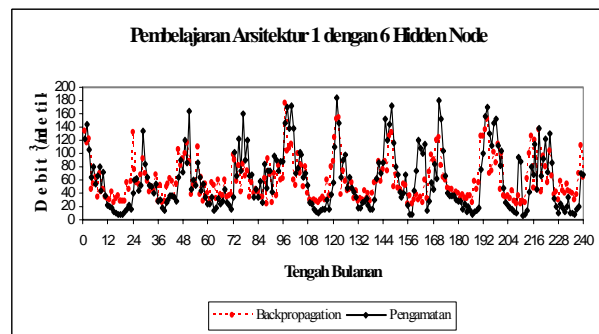
**Tabel 1. Parameter statistika hasil pembelajaran metoda *backpropagation***

No	Arsitek-tur	KAR	MSE	Stadar deviasi	Koef. korelasi
1	A1-4	0.6816	0.0012	31.2537	0.7312
2	A1-6	0.6876	0.0011	31.1649	0.7305
3	A2-4	0.6605	0.0241	31.5617	0.7119
4	A2-6	0.6576	0.0235	31.6037	0.7116
5	A3-4	0.4872	0.0005	35.8629	0.8634
6	A3-6	0.4702	0.0005	36.2256	0.8669
7	A4-4	0.3136	0.0082	38.7818	0.8794
8	A4-6	0.3136	0.0083	38.7998	0.8785

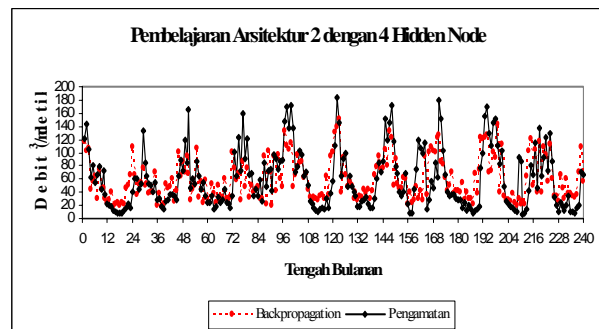
Hidrograf hasil pembelajaran dengan metoda backpropagation disajikan pada **Gambar 6** sampai **13**.



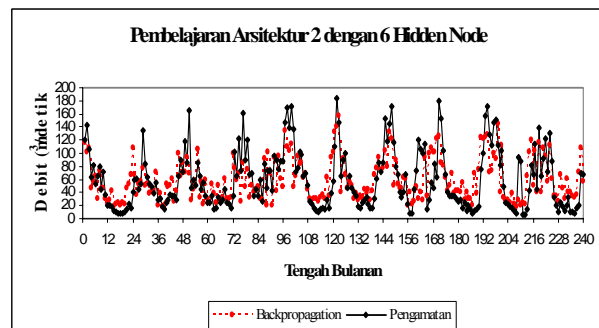
**Gambar 6. Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 1 dengan 4 *hidden node***



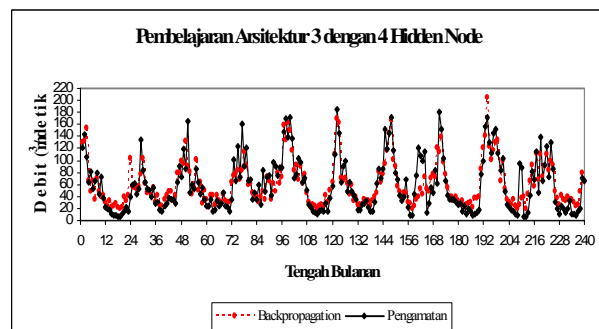
**Gambar 7. Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 1 dengan 6 *hidden node***



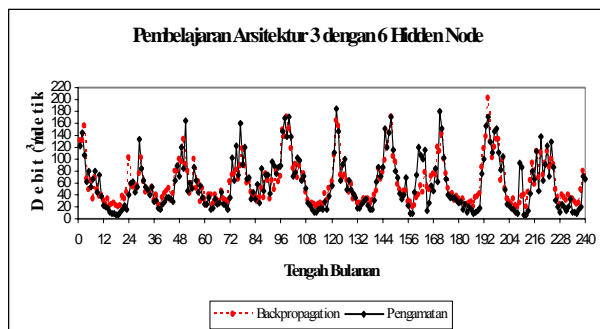
**Gambar 8. Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 2 dengan 4 *hidden node***



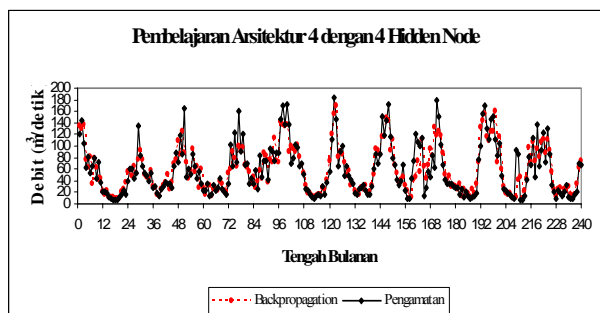
**Gambar 9. Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 2 dengan 6 *hidden node***



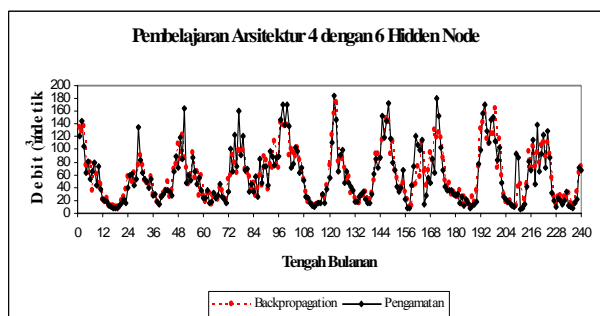
**Gambar 10. Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 3 dengan 4 *hidden node***



**Gambar 11.** Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 3 dengan 6 *hidden node*



**Gambar 12.** Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 4 dengan 4 *hidden node*



**Gambar 13.** Hidrograf hasil pembelajaran metoda *backpropagation* arsitektur 4 dengan 6 *hidden node*

**Tabel 2.** Parmeter statistika hasil pengujian

No	Pola	KAR	MSE	Data Max	Data Min	Data Rata-rata	Standar Deviasi	Koefisien Skewnes	Koefisien Korelasi
1	A1-4	0.836	0.004	207.132	11.967	51.425	34.243	1.464	0.735
2	A1-6	0.828	0.004	220.959	11.504	50.896	34.511	1.654	0.735
3	A2-4	0.663	0.013	189.168	6.793	48.563	39.736	1.426	0.751
4	A2-6	0.681	0.013	188.896	5.626	49.560	39.360	1.307	0.754
5	A3-4	0.546	0.003	198.253	7.770	52.214	43.775	1.497	0.805
6	A3-6	0.512	0.004	270.208	6.893	49.698	46.518	2.209	0.751
7	A4-4	0.451	0.008	171.108	6.480	48.882	41.564	1.275	0.813
8	A4-6	0.447	0.008	167.197	6.424	48.768	41.301	1.252	0.813
Pengamatan				301.900	4.000	47.108	46.146	2.212	

Dari proses pembelajaran dihasilkan bobot hubungan, bobot-bobot tersebut kemudian digunakan untuk membangkitkan data sintetik atau disebut pengujian. Hasil data bangkitan dihitung KAR-nya, kemudian dipilih bobot yang menghasilkan data bangkitan yang memiliki KAR terkecil untuk setiap arsitektur, kemudian dihitung parameter statistiknya. Proses pengujian menghasilkan parameter statistika untuk masing-masing pola arsitektur seperti pada **Tabel 2**.

Dari **Tabel 2** diketahui bahwa arsitektur 4 dengan 6 *hidden node* (A4-6) memiliki nilai KAR dan MSE paling kecil yaitu 0,447 dan memiliki koefisien korelasi paling besar yaitu 0,813 atau 81,3%. Sehingga disimpulkan arsitektur 4 dengan 6 *hidden node* (A4-6) menghasilkan data bangkitan yang paling optimum pada proses pengujian. Hasil pengujian untuk setiap pola arsitektur dalam bentuk hidrograf disajikan pada **Gambar 14** sampai **21**, sedangkan dalam bentuk *flow duration curve* disajikan dalam **Gambar 22** sampai **29**.

## 10. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasan dapat diambil kesimpulan sebagai berikut :

1. Penerapan metode *backpropagation* untuk pemodelan curah hujan limpasan memberikan hasil yang relatif baik pada proses pembelajaran dan proses pengujian.
2. Arsitektur 4 dengan 6 *hidden node* (A4-6) memiliki nilai KAR paling kecil yaitu 0,447 dan koefisien korelasi paling besar yaitu 0,813. Sehingga disimpulkan arsitektur 4 dengan 6 *hidden node* (A4-6) menghasilkan data bangkitan yang paling optimum pada proses pengujian.
3. Secara umum ANN dapat diterapkan dalam pemodelan curah hujan limpasan walaupun hasilnya tidak terlalu akurat karena masih terjadi penyimpangan.

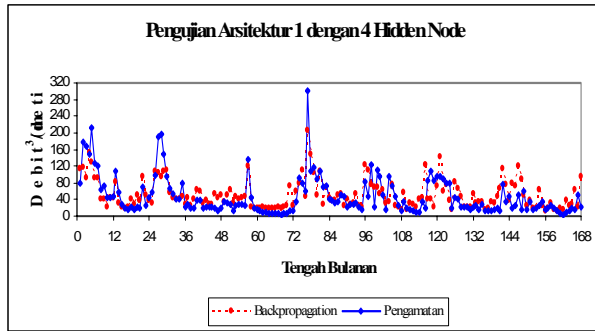
4. Adanya penyimpangan data hasil pemodelan dengan data hasil pengamatan mungkin disebabkan adanya hal yang tidak diperhitungkan dalam pemodelan ini seperti pengambilan air pada sungai yang menyebabkan berubahnya kondisi alamiah aliran sungai tersebut serta kualitas data yang ada.

Guna mendapatkan hasil yang lebih akurat lagi, perlu dikembangkan pemodelan dengan menggunakan arsitektur dan metode pembelajaran yang barangkali lebih sesuai dengan permasalahan ini, misalnya dengan teknik optimasi (*non traditional*) dalam pembobotan ANN menggunakan *genetic algorithm*. Di samping itu perlu diperhatikan kualitas data yang ada dan perlu diperhitungkan adanya pengambilan air yang menyebabkan berubahnya kondisi alamiah sungai .

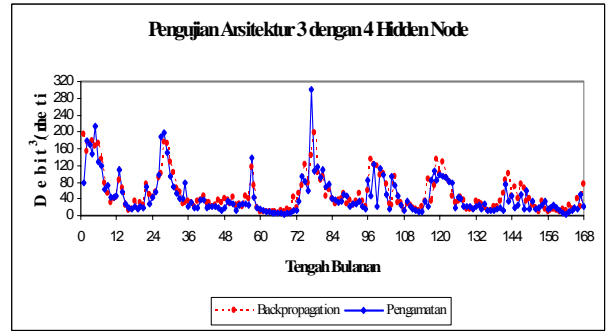
### **Daftar Pustaka**

- Adidarma, W.K., Hadihardaja, I.K., Legowo, S., 2004, "*Perbandingan Pemodelan Hujan-Limpasan Antara Artificial Neural Network (ANN) dan NRECA*", Jurnal Teknik Sipil ITB, Vol. 11 No. 3, 105 – 115.
- Ferianto, S.D., Hadihardaja, I.K., 2003, "*Pemodelan Multivariat Deret Waktu Sumberdaya Air Menggunakan Jaringan Saraf Buatan*", Jurnal Pengembangan Keairan Badan Penerbit Undip, 1 Tahun 10, Juli 2003, 58 – 75.
- Hadihardaja, I.K., Sutikno, S., 2005, "*Aplikasi Metode Generalized Reduced Gradient dalam Pemodelan Curah Hujan-Limpasan Menggunakan Artificial Neural Network*", Jurnal Media Komunikasi Teknik Sipil, Juni.
- Hadihardaja, I.K., "*Stream Flow Discharge and Relation Using Artificial Neural Network*", Jurnal Media Komunikasi Teknik Sipil, 10, 1 – 15.
- Hadihardaja, I.K., 2003, "*Model Pengoperasian Waduk Tunggal dengan Jaringan Syaraf Tiruan*", Jurnal Pengembangan Keairan Badan Penerbit Undip, 2 Tahun 10, 2003 , 24 – 33.
- Laurence, F., 1994, "*Fundamental of Neural Networks*", Prentice Hall, Englewood cliffs, New Jersey.
- Simon, H., 1994, "*Neural Networks, A Comprehensive Foundation*", Macmillan College Publishing Company, USA.
- Sri, K., 2003, "*Artificial Intelligence (Teknik dan Aplikasinya)*", Graha Ilmu.
- Vladan, B., Bojkov, H., Ventzi, 2001, "*Runoff Modelling with Genetic Programming and Artificial Neural Networks*", D2K Technical Report, D2K TR 0401-1, April.
- Vladan, B., Gopakumar, 1999, "*Seabed Recognition Using Neural Networks*", D2K Technical Report, D2K TR 0399-1, Marc.

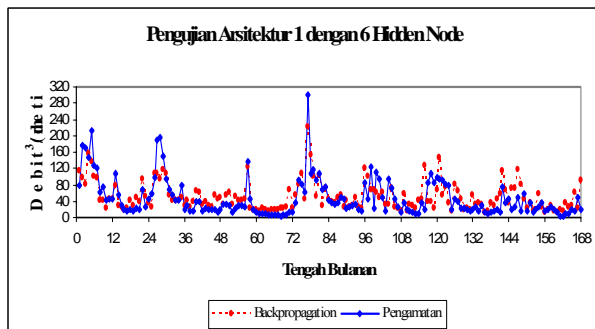




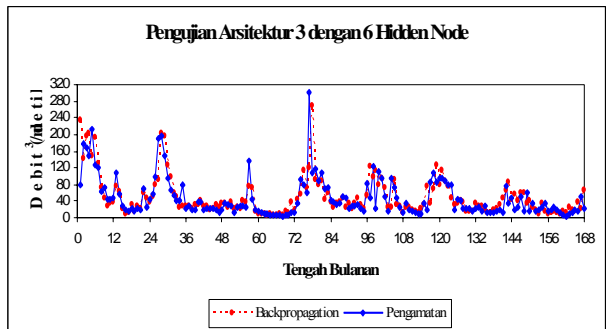
Gambar 14. Hidrograf hasil pengujian arsitektur 1 dengan 4 *hidden node*



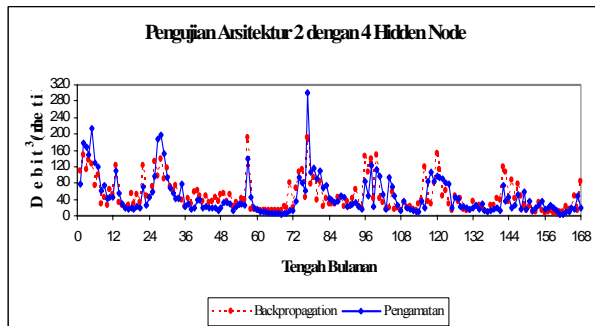
Gambar 18. Hidrograf hasil pengujian arsitektur 3 dengan 4 *hidden node*



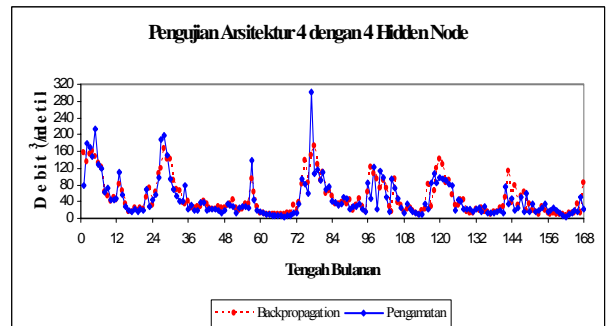
Gambar 15. Hidrograf hasil pengujian Arsitektur 1 dengan 6 *hidden node*



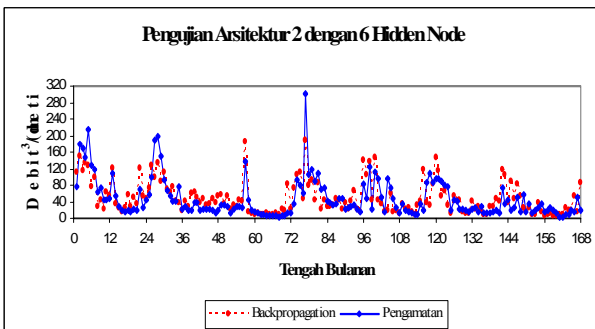
Gambar 19. Hidrograf hasil pengujian arsitektur 3 dengan 6 *hidden node*



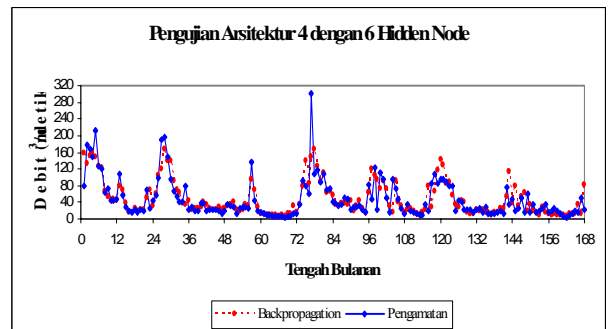
Gambar 16. Hidrograf hasil pengujian arsitektur 2 dengan 4 *hidden node*



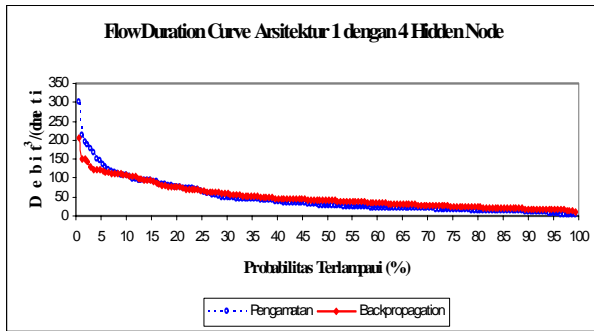
Gambar 20. Hidrograf hasil pengujian arsitektur 4 dengan 4 *hidden node*



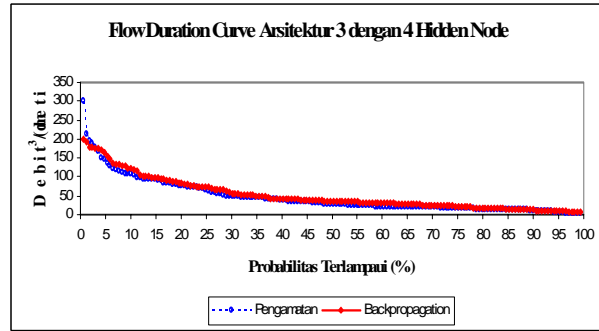
Gambar 17. Hidrograf hasil pengujian arsitektur 2 dengan 6 *hidden node*



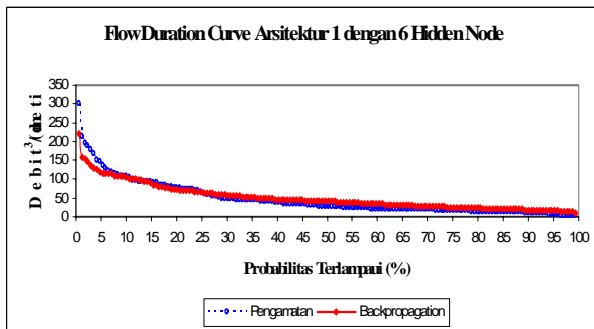
Gambar 21. Hidrograf hasil pengujian arsitektur 4 dengan 6 *hidden node*



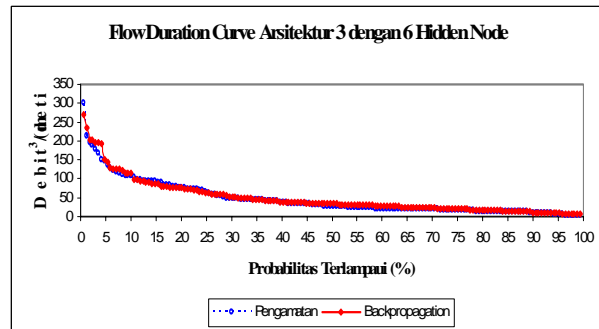
Gambar 22. Flow duration curve arsitektur 1 dengan 4 hidden node



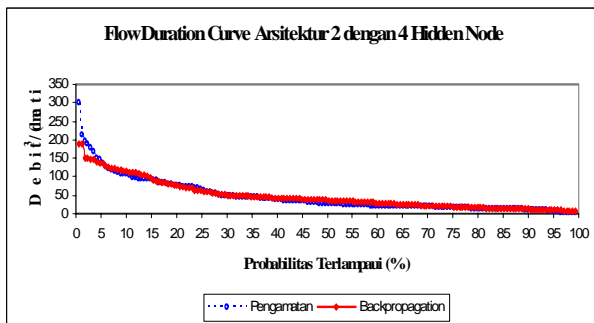
Gambar 26. Flow duration curve arsitektur 3 dengan 4 hidden node



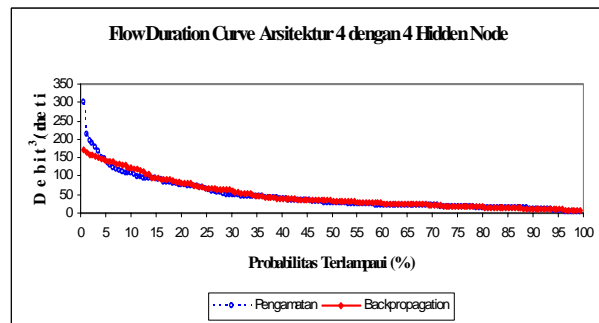
Gambar 23. Flow duration curve arsitektur 1 dengan 6 hidden node



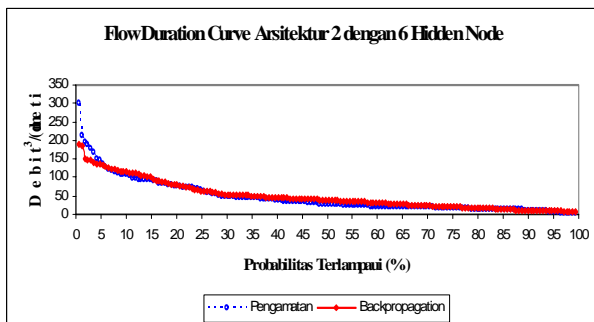
Gambar 27. Flow duration curve arsitektur 3 dengan 6 hidden node



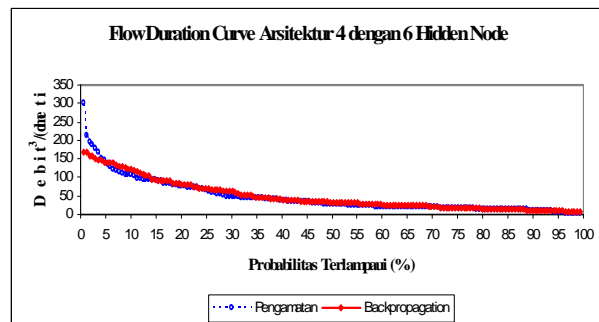
Gambar 24. Flow duration curve arsitektur 2 dengan 4 hidden node



Gambar 28. Flow duration curve arsitektur 4 dengan 4 hidden node



Gambar 25. Flow duration curve arsitektur 2 dengan 6 hidden node



Gambar 29. Flow duration curve arsitektur 4 dengan 6 hidden node